

# Treball de Fi de Màster

Màster Universitari en Enginyeria Industrial (MUEI)

## Segmentació de text manuscrit amb la SVD

### MEMÒRIA

19 de setembre de 2019

**Autor:** Sergi Panisello Barberà

**Director:** Jaume Amorós Torrent

**Convocatòria:** 09/2019



Escola Tècnica Superior  
d'Enginyeria Industrial de Barcelona



# ETSEIB



## Resum

El treball present continua la investigació començada per la Judith Bergadà i posteriorment el José de la Puente davant de resultats esperançadors de reconeixement de text mitjançant la descomposició en valors singulars (SVD) aplicat a lletres separades entre sí i amb espais equidistants.

La continuació d'aquest projecte consisteix en donar un pas més en l'aplicació d'aquesta metodologia i eliminar aquestes restriccions imposades anteriorment per intentar reconèixer lletra lligada.

En primer lloc, el contingut d'aquest treball consisteix en una introducció a la SVD. Posteriorment, es determinen els invariants derivats d'aquesta descomposició que serviran per elaborar un catàleg patró. El text a reconèixer es compararà amb aquest catàleg.

En segon lloc, s'explicarà l'algoritme aplicat per a la creació d'aquest catàleg patró i l'algoritme dissenyat per al reconeixement de text manuscrit amb lletra lligada.

Donat que en aquest projecte la segmentació es realitza a nivell de paraula, no es possible determinar la taxa de reconeixement realitzant una comparació lletra a lletra ja que és possible que la longitud de la paraula reconeguda i de la paraula escrita no sigui igual. És per aquest motiu que es defineix la distància de Levenshtein com a criteri per definir la taxa d'encerts.

Donat el baix percentatge d'èxit de l'algoritme dissenyat, s'expliquen les millores que s'han introduït per reduir la distància entre el text reconegut i escrit.

Finalment es presenten els resultats obtinguts on s'aconsegueix una taxa de reconeixement al voltant del 70 %. Resultats molt satisfactoris ja que s'ha obtingut un augment del 50% en la taxa d'encert des del primer algoritme funcional aplicat.

Donat els resultats s'aconsella continuar endavant aquest projecte per incloure signes de puntuació, accents i augmentar la velocitat de càlcul.

Finalment, s'explica la implementació del codi i s'aborden altres aspectes de caire diferent al algoritme en sí, com el pressupost del projecte i l'estudi de l'impacte ambiental d'aquest.



# Índex

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Prefaci</b>   | <b>5</b>  |
| 1.1      | Origen del projecte                                    | 5         |
| 1.2      | Motivació  | 5         |
| 1.3      | Requeriments previs                                    | 5         |
| <b>2</b> | <b>Introducció</b>                                     | <b>7</b>  |
| 2.1      | Objectius del projecte                                 | 7         |
| 2.2      | Abast del projecte                                     | 7         |
| 2.3      | Estat de l'art   | 7         |
| <b>3</b> | <b>Introducció al reconeixement òptic de caràcters</b> | <b>9</b>  |
| 3.1      | Definició i origen                                     | 9         |
| 3.2      | Variants del reconeixement òptic                       | 9         |
| <b>4</b> | <b>Fonaments teòrics</b>                               | <b>11</b> |
| 4.1      | Valors singulars                                       | 11        |
| 4.2      | Descomposició en valors singulars                      | 11        |
| 4.3      | Expansió SVD   | 12        |
| <b>5</b> | <b>L'expansió SVD en el processament d'imatges</b>     | <b>13</b> |
| 5.1      | Invariants   | 14        |
| 5.1.1    | Canvis d'escala en l'eix X                             | 14        |
| 5.1.2    | Canvis d'escala en l'eix Y                             | 15        |
| 5.1.3    | Components a estudiar                                  | 15        |
| <b>6</b> | <b>Catàleg</b>   | <b>17</b> |
| 6.1      | Creació del catàleg                                    | 17        |
| 6.1.1    | Correcció dels reflexos                                | 18        |
| 6.1.2    | Binaritzat   | 18        |
| 6.1.3    | Segmentació per lletres                                | 19        |
| 6.1.4    | Obtenció de les components                             | 20        |
| <b>7</b> | <b>Reconeixement de lletra manuscrita</b>              | <b>21</b> |
| 7.1      | Segmentació de lletres                                 | 21        |
| 7.1.1    | Cerca iterativa de lletra candidata                    | 22        |
| 7.1.2    | Comparació de la candidata amb el catàleg              | 24        |
| 7.1.3    | Elecció de la millor candidata                         | 25        |
| 7.2      | Transcripció a text digital                            | 25        |
| <b>8</b> | <b>Millora de la taxa d'encert</b>                     | <b>27</b> |
| 8.1      | Avaluació de la taxa d'encert                          | 27        |
| 8.2      | Creació d'un catàleg representatiu de lletra lligada   | 27        |
| 8.2.1    | Eliminació de valors atípics                           | 29        |
| 8.2.2    | Catàleg segons la freqüència d'aparició de la lletra   | 30        |
| 8.3      | Reducció de lletres candidates                         | 32        |
| 8.4      | Elecció del millor catàleg i resultats obtinguts       | 34        |
| <b>9</b> | <b>Implementació del codi</b>                          | <b>37</b> |

|           |   |           |
|-----------|---|-----------|
| 9.1       | Codi per a la creació del catàleg . . . . . | 37        |
| 9.2       | Codi per al reconeixement . . . . .         | 37        |
| <b>10</b> | <b>Pressupost</b>                           | <b>39</b> |
| <b>11</b> | <b>Impacte ambiental</b>                    | <b>41</b> |
|           | <b>Conclusions</b>                          | <b>43</b> |
|           | <b>Agraïments</b>                           | <b>45</b> |
|           | <b>Bibliografia</b>                         | <b>45</b> |

## Índex de figures

|    |   |    |
|----|---|----|
| 1  | Lletra reconstruïda amb diferent nombre de valors singulars . . . . .       | 13 |
| 2  | Funció dividida en $k$ parts entre 0 i 1 . . . . .                          | 16 |
| 3  | Exemple plantilla catàleg . . . . .   | 17 |
| 4  | Procés correcció reflexos escaneig . . . . .                                | 18 |
| 5  | Procés de binaritzat . . . . .  | 19 |
| 6  | Delimitació de la línia . . . . .   | 19 |
| 7  | Delimitació de la lletra . . . . .  | 19 |
| 8  | Segmentació de lletres dins d'una paraula . . . . .                         | 22 |
| 9  | Diferents tipus de mallat . . . . .   | 22 |
| 10 | Candidates a lletra . . . . .   | 23 |
| 11 | Procés de búsqueda de candidates . . . . .                                  | 23 |
| 12 | Paraula amb separació vertical . . . . .                                    | 25 |
| 13 | Distribució espais exclouent valors menors de 50 píxels . . . . .           | 26 |
| 14 | Diferència lletra lligada amb lletra separada . . . . .                     | 28 |
| 15 | Lletres amb diferents condicions de contorn . . . . .                       | 29 |
| 16 | Lletres que provoquen errors . . . . .                                      | 30 |
| 17 | Distribució altura lletres . . . . .  | 33 |
| 18 | Nou procés de búsqueda de candidates . . . . .                              | 34 |
| 19 | Text a reconèixer . . . . .   | 35 |
| 20 | Procés d'encament d'errors de la paraula was . . . . .                      | 35 |
| 21 | Alguns errors ordenats en l'ordre en el qual apareixen en el text . . . . . | 36 |

# 1 Prefaci

## 1.1 Origen del projecte

Uns dels aspectes que caracteritzen els humans és el llenguatge i la seva comunicació. El fil conductor del llenguatge és la escriptura com a mitja de representació a través de signes i traços per tal de conservar la informació i transmetre-la.

En els últims temps, no es estrany sentir a parlar de termes com la intel·ligència artificial, en el qual un ordinador o en el seu defecte una màquina és capaç de realitzar accions cognitives que els humans associen amb la ment humana.

En aquest treball s'exploraran les vies de reconeixement de text manuscrit, les quals s'han donat a conèixer i s'han obtingut cada cop millors resultats, tot i així, s'utilitzarà un mètode poc convencional als mètodes usualment aplicats per al reconeixement de text, és a dir, mitjançant la descomposició en valors singulars (SVD).

L'origen del treball recau al tutor del projecte el qual prèviament ha tutoritzat dos treballs relacionats amb aquesta temàtica. El primer, el 2016 desenvolupat per Berguedà en el qual es va adaptar el reconeixement òptic de caràcters (OCR) per basar-lo amb la SVD [1]. En aquest projecte es va investigar la viabilitat de reconèixer text tipografiat de 3 alfabetos diferents com el llatí, el rus i el grec. Donat els bons resultats d'aquest projecte el 2017, es realitza un altre treball relacionat amb el reconeixement de text mitjançant la SVD a càrrec del José de la Fuente [2], el qual anava un pas endavant en el reconeixement de text, ja que en aquest es comprovà la viabilitat del mètode per poder reconèixer text manuscrit, és a dir, escrit a mà i no imprès amb lletres distanciades entre sí. Donat els bons resultats del José, vaig tenir l'oportunitat de poder continuar aquest estudi.

Així doncs, el projecte actual és la continuació dels projectes de la Judith Berguedà i del José de la Puente, en el qual es continuarà desenvolupant aquesta metodologia per reconèixer text manuscrit.

## 1.2 Motivació

La motivació per dur a terme aquest treball és continuar amb l'investigació duta a terme anteriorment, i poder avançar més en el reconeixement de text manuscrit, més conegut per les seves sigles de l'anglès HTR (*handwritten text recognition*).

El projecte del José de la Puente consisteix en identificar una lletra manuscrita que estigui aïllada, és a dir que estigui a certa distància de qualsevol altra lletra per poder indentificar-la. En aquest projecte, es partirà de text escrit junt amb lletra lligada, el qual s'haurà de segmentar, aïllant cada lletra per separat i fer el reconeixement de la lletra.

Degut a l'èxit del projecte desenvolupat pel José de la Puente, es vol investigar si es serà capaç de segmentar el text i reconèixer les lletres contingudes.

## 1.3 Requeriments previs

Els requeriments puntals per a desenvolupar aquest projecte són el tenir coneixement d'àlgebra lineal, així com també programació amb MATLAB.

L'àlgebra lineal és necessària per dur a terme l'algoritme de reconeixement. Concretament, la descomposició en valors singulars ja que és el cor des del qual es du a terme aquest treball.

D'altra banda, MATLAB és l'eina utilitzada per realitzar l'implementació de l'algoritme. Per tant, tenir un coneixement avançat de programació en el programa és un dels requisits.

L'elecció d'aquest programari ve determinada perquè és d'elevada utilitat en l'entorn d'aquest projecte en el qual s'han de realitzar diverses proves. També s'ha de tenir en compte que aquest programari es disposa llicència gratuïta d'estudiant proporcionada per la UPC.

Un altre requisit és disposar de la informació aportada per projecte anterior (José de la Puente, 2017) ja que així es parteix d'una base a partir de la qual edificar aquest nou projecte.



## 2 Introducció

### 2.1 Objectius del projecte

L'objectiu d'aquest projecte es a partir del estudis previs implementar la descomposició SVD per tal de detectar text manuscrit no aïllat. És a dir, a partir d'una pàgina escrita de forma manual reconèixer les paraules lletra a lletra, sense haver d'aïllar cada lletra per separat.

Amb aquest objectiu, es tractarà de:

- Entendre l'algoritme realitzat en el projecte anterior per poder-hi treballar.
- Ser capaç d'aconseguir una taxa d'encert de caràcters superior al 40% a l'hora de reconèixer lletra lligada.
- Identificar possibles errors per poder millorar el procés de reconeixement.

### 2.2 Abast del projecte

En aquest projecte es consideraran els textos que compleixin les característiques següents:

- Línies horitzontals ( $\pm 1^\circ$ ) separades de forma que una recta horitzontal només ha d'impactar amb una línia.
- No es detectaran accents ni signes de puntuació.
- Les lletres no tindran un solapament d'una amb l'altra, cada lletra ha d'estar continguda dintre de dos rectes verticals. Per exemple, el pal de la t no hauria d'invadir l'espai de la lletra anterior o posterior.

Tot i aquestes restriccions es poden crear catàlegs que incloguin símbols, altres alfabetes, etc. L'algoritme no es limita a l'ús descrit en aquest treball.

### 2.3 Estat de l'art

Actualment s'utilitzen diferents algoritmes per al reconeixement de text escrit a mà: algoritmes estadístics, algoritmes estructurals, algoritmes amb xarxes neuronals i màquines de vector suport. [3]

Les xarxes neuronals és una metodologia molt de moda en l'actualitat. Tot i això, el *deep learning*, també guanya detractors i és que a aquests mètodes se'ls dona molt bé trobar patrons en les dades, però aquest enfocament no revela la causa-efecte i resulta molt limitat per a entendre el món real. [4]

Cal remarcar que normalment no s'aplica un sol algoritme per determinar el contingut d'un text, sinó que s'executen en paral·lel per aconseguir major èxit en el reconeixement. També es usual determinar la paraula final de manera que sigui la més propera a un diccionari de paraules preestablert.



### 3 Introducció al reconeixement òptic de caràcters

Actualment, la visió per computador és una disciplina científica d'elevada importància en el món actual. És habitual ja sentir a parlar de reconeixement facial, del cotxe autònom, *smart cities*, etc.

Totes aquestes tecnologies no serien possibles sense aquesta branca de la enginyeria, i és que la seua aplicabilitat és molt més important del que un es pot arribar a imaginar. Per exemple:

- Processos industrials: Control de qualitat, del procés productiu, etc.
- Medi ambient i energia: Gestió de residus, control de frau (explotacions o construccions il·legals), agricultura (estat del terreny agrícola, alertes de poda), instal·lacions (fugues de calor o gasos), desastres naturals, etc.
- *Smart cities*: Moviment urbà (averies en via pública, control infraestructura de transport), mobilitat (estat del trànsit, gestió aparcaments), urbanisme (mapes 3D intel·ligents, eficiència energètica edificis), etc.
- *Retail*: Logística (control estat del magatzem), superfícies comercials (control flux clients i tendències), etc.
- Banca i assegurances: Gestió documental (classificació de documents), peritatges (peritatges de forma automàtica), etc.

En el cas del reconeixement de caràcters, la creixent tendència en l'obtenció de dades, donat l'auge del *big data*, és clarament creixent. Guardar la informació ja sigui escrita o manuscrita en forma d'imatges comporta una despesa en espai important. Extreure la informació d'aquest text i gestionar-la de forma eficient és de vital importància.

#### 3.1 Definició i origen

El reconeixement òptic de caràcters o més conegut com OCR degut a la nomenclatura anglesa (*Optical Character Recognition*) és el procés que partint d'una imatge obtinguda a través de la digitalització d'un text, s'identifica de forma automàtica els caràcters o símbols continguts que pertanyen a un alfabet específic. Amb l'objectiu de desar el contingut d'aquest text en forma de dades.

La història del reconeixement òptic de caràcters comença abans de la primera guerra mundial [5] ja que en els antics estudis de reconeixement de patrons, la majoria de gent empenia projectes d'OCR. Una de les raons era perquè tractar amb caràcters semblava fàcil i es considerava un problema que es podria resoldre fàcilment.

Tot i això després d'un temps de progrés van aparéixer els primers problemes amb l'OCR, fet que va diversificar als investigadors en diferents tòpics com visió per computador, reconeixement d'objectes 3-D, etc. [6]

#### 3.2 Variants del reconeixement òptic

Actualment, segons la informació de partida es poden separar les tecnologies en tres tipus:

- OMR (*optical mark recognition*): Sistema que detecta presència o no de marques per tal de decidir la lletra. Usat típicament en exàmens, a partir d'una plantilla preestablerta.
- ICR (*intelligent character recognition*): Es parteix d'una plantilla en la qual els caràcters s'han descriure dintre dels requadres, delimitant on es situarà cada caràcter. Aquest sistema permet reconèixer text manuscrit, usualment en majúscules.
- OCR (*optical character recognition*): Reconeix imatges de text imprès, de forma que escanejant qualsevol pàgina escrita, transcriu el text que conté.
- HTR (*handwritten text recognition*): Aquest tipus de reconeixement pot ser de dues maneres diferents. La primera és escrivint a través d'una pantalla tàctil en la qual s'enregistren els moviments que fa el l'escriptor (*online*). La segona és tracta de reconèixer de la mateixa forma que l'OCR, però en aquest cas partint de text manuscrit (*offline*).

En el cas d'aquest projecte la tecnologia és la HTR *offline*, a partir d'una imatge manuscrita, es vol obtenir la informació del text amb dades, de forma que la taxa d'encert sigui màxima.

## 4 Fonaments teòrics

Qualsevol matriu  $A \in \mathbb{R}^{m \times n}$ , té una factorització de la forma:  $A = U \cdot \Sigma \cdot V^t$ . On  $U \in \mathbb{R}^{m \times n}$  en columnes ortogonals, i  $V \in \mathbb{R}^{n \times n}$  i una matriu  $\Sigma \in \mathbb{R}^{n \times n}$  diagonal.

$$\Sigma = \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{pmatrix}$$

Per trobar aquesta factorització en valors singulars, s'ha de veure el següent.

### 4.1 Valors singulars

Per a una matriu  $A \in \mathbb{R}^{m \times n}$ , la matriu  $A^t A$  és simètrica i per tant, pot ser diagonalitzada ortogonalment. Els valors propis de  $A^t A$  són reals i no negatius. Com es pot veure en la equació 1, la imatge del vector propi és igual al producte del valor propi  $\lambda$  pel vector propi  $u$ . Tenint:

$$A^t A u = \lambda u \quad (1)$$

Multiplicant als dos costats es manté l'igualtat, resultant:

$$u^t A^t A u = \lambda u^t u \quad (2)$$

que indica:

$$\|A u\|^2 = \lambda \|u\|^2 \implies \lambda > 0 \quad (3)$$

#### Definició:

Si  $A$  és una matriu  $m \times n$ , els valors singulars de  $A$  són les arrels quadrades dels valors propis de  $A^t A$ , i es denoten com  $\sigma_1, \dots, \sigma_n$ . És a dir,  $\lambda_i = \sqrt{\sigma_i}$ .

### 4.2 Descomposició en valors singulars

Considerant  $A$  la matriu d'una aplicació lineal  $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$  existeix una descomposició  $A = U \cdot \Sigma \cdot V^t$  en bases canòniques de sortida i arribada, amb  $U \in \mathbb{R}^{m \times m}$ ,  $V \in \mathbb{R}^{n \times n}$  i

$$\Sigma = \begin{pmatrix} D & O \\ O & O \end{pmatrix}$$

$$D = \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{pmatrix}$$

tals que  $\Sigma$  és una matriu de  $f$  en bases ortonormals  $v_1, \dots, v_n$  de sortida i  $u_1, \dots, u_n$  d'arribada.

On  $O$  són les matrius pertinents nul·les, i per tant,  $\sigma_{r+1} = \sigma_{r+2} = \sigma_{r+3} = 0$ .  $\Sigma$  és una matriu en base ortonormal.

### Teorema

1.  $(A \cdot v_1, \dots, A \cdot v_r)$  és el conjunt ortogonal.
2.  $\left(\frac{A \cdot v_1}{\sigma_1}, \dots, \frac{A \cdot v_r}{\sigma_r}\right)$  és una base ortonormal del subespai  $Col(A)$ .
3.  $(v_{r+1}, \dots, v_n)$  és una base ortonormal del subespai  $Nul(A)$ .
4.  $rang(A) = r$  coincideix amb el nombre de valors singulars no nuls.

### 4.3 Expansió SVD

Mitjançant aquesta descomposició, es pot veure que:

$$A = \sum_{i=1}^r \sigma_i u_i v_i^T \quad (4)$$

Demostració:

$$A = U \cdot \Sigma \cdot V^t = (u_1, \dots, u_r, \dots, u_n) \begin{pmatrix} D & O \\ O & O \end{pmatrix} \begin{pmatrix} v_1^T \\ \vdots \\ v_r^T \\ \vdots \\ v_k^T \end{pmatrix}; \quad n, k > r \quad (5)$$

$$A = (u_1, \dots, u_r, \dots, u_n) \begin{pmatrix} D \\ \vdots \\ v_r^T \end{pmatrix} = (u_1, \dots, u_r) \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{pmatrix} \begin{pmatrix} v_1^T \\ \vdots \\ v_r^T \end{pmatrix} = \sum_{i=1}^r \sigma_i u_i v_i^T \quad (6)$$

## 5 L'expansió SVD en el processament d'imatges

Una imatge no deixa de ser una matriu en el qual els seus índex indiquen la informació necessària per representar un píxel. En aquest treball, es tractaran en imatges en blanc i negre; i cada índex de la matriu indicarà l'escala de color de blanc fins a negre. Hi ha diversos tipus de variables per representar-ho. Per exemple, el tipus de variable *double*, escala la intensitat de gris des de 0 fins a 1. Sent 0 l'absència de color, és a dir, negre i 1 blanc.

Tractar amb tota la informació continguda en una imatge és molt complicat i requereix un esforç computacional enorme. Per tractar una imatge s'ha de comprimir de forma eficient sense perdre informació rellevant.

Un cop fet aquesta petita introducció i haver explicat la SVD, s'explica l'importància d'aquesta descomposició. Realitzant-la a una imatge, es pot guardar la seva informació tenint en compte els valors singulars més importants, és a dir, els més grans.

És a dir, si es tenen els valors singulars ordenats de major a menor tal que  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r$ . Una aproximació de la imatge es pot obtenir utilitzant l'equació 6, amb el sumatori fins a  $n$  on  $n < r$ , com per exemple  $n = 3$ :

$$\sum_{i=1}^3 \sigma_i u_i v_i^T = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \sigma_3 u_3 v_3^T$$

Reduint el contingut de les dades considerablement.

Un exemple es pot veure fent la SVD de la lletra d. Com es pot veure a la figura 1, com més valors singulars es prenen més nítida és la imatge.

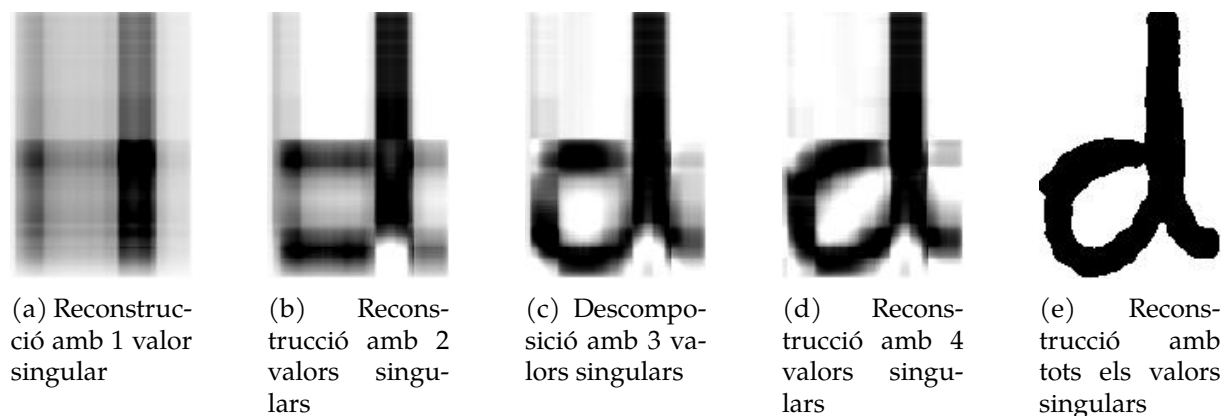


Figura 1: Lletra reconstruïda amb diferent nombre de valors singulars

Un altre aspecte important són els vectors singulars, ja que en text tipografiat hi ha lletres que comparteixen valors singulars com la "b", "p", "q" i "d" degut a les seves simetries. És a dir, la diferència entre aquests valors singulars és de l'ordre de  $10^{-11} \%$ . L'única manera per poder diferenciar-les és amb els vectors propis. Tot i així, remarcar que en l'àmbit d'aquest projecte de text manuscrit difereixen lleugerament (aproximadament un 10%) i depenent molt del tipus de cal·ligrafia que usi l'escriptor.

Un cop ja se sap comprimir una imatge gràcies a la SVD, el següent pas és com es tracta aquesta

informació per comparar imatges que tenen diferents propietats.

## 5.1 Invariants

Per tal que una lletra sigui comparable a una altra és interessant que es facin amb paràmetres invariants. És a dir que si es fa una lletra més aplatada o més gran no afecti a l'algoritme.

És per això, que s'han d'obtenir les propietats necessàries de la descomposició per tal que sigui invariant a aquests canvis.

### 5.1.1 Canvis d'escala en l'eix X

Donada una matriu  $M$  que representa qualsevol caràcter, es pot obtenir la matriu  $\tilde{M}$  = obtinguda del canvi d'escala en l'eix OX a través del producte Kronecker, de forma que el mateix píxel estigui repetit  $n$  vegades.  $\tilde{M} = M \otimes (1, 1, \dots, 1_n)$  Exemple:

$$M = \begin{pmatrix} 1 & 1 & \dots \\ 1 & 0 & \dots \\ 0 & \ddots & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

$$\tilde{M} = M \otimes (1, 1, \dots, 1_n) =$$

$$\begin{pmatrix} 1 \begin{pmatrix} 1 & \dots & 1_n \end{pmatrix} & 1 \begin{pmatrix} 1 & \dots & 1_n \end{pmatrix} & \dots \\ 1 \begin{pmatrix} 1 & \dots & 1_n \end{pmatrix} & 0 \begin{pmatrix} 1 & \dots & 1_n \end{pmatrix} & \dots \\ 0 \begin{pmatrix} 1 & \dots & 1_n \end{pmatrix} & \ddots & \dots \\ \vdots & \vdots & \ddots \end{pmatrix} = \begin{pmatrix} \begin{pmatrix} 1 & \dots & 1_n \end{pmatrix} & \begin{pmatrix} 1 & \dots & 1_n \end{pmatrix} & \dots \\ \begin{pmatrix} 1 & \dots & 1_n \end{pmatrix} & \begin{pmatrix} 0 & \dots & 0_n \end{pmatrix} & \dots \\ \begin{pmatrix} 0 & \dots & 0_n \end{pmatrix} & \ddots & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

La relació entre les matrius  $\tilde{M}$  i  $M$  és:

1. Els valors singulars no nuls de  $\tilde{M}$  són els de  $M$  multiplicats per  $\sqrt{n}$ .
2. Si  $\sigma \neq 0$  és un valor singular de  $M$ , amb vector de sortida  $v = (v_1, v_2, \dots)$ . El valor singular  $\sigma_{\tilde{M}} = \sigma_M \sqrt{n}$  té vector de sortida

$$\tilde{v} = \left( \frac{v_{11_1}}{\sqrt{n}}, \dots, \frac{v_{11_n}}{\sqrt{n}}, \frac{v_{12_1}}{\sqrt{n}}, \dots, \frac{v_{12_n}}{\sqrt{n}}, \dots, \frac{v_{21_1}}{\sqrt{n}}, \dots, \frac{v_{21_n}}{\sqrt{n}}, \frac{v_{22_1}}{\sqrt{n}}, \dots, \frac{v_{22_n}}{\sqrt{n}}, \dots \right).$$

Cada component de cada vector  $v_i$  de sortida es repeteix  $n$  vegades dividit per  $\sqrt{n}$ .

3. Si  $\sigma \neq 0$  és un valor singular de  $M$ , el vector singular d'arribada associat a aquest vector és igual al vector propi de  $\tilde{M}$ . És a dir,  $u_i = \tilde{u}_i$



### 5.1.2 Canvis d'escala en l'eix Y

Es pot fer la mateixa analogia, ara aplicada a l'eix Y. Donada una matriu  $M$  que representa qualsevol caràcter, es pot obtenir la matriu  $\tilde{M}$  = obtinguda del canvi d'escala en l'eix OY a través del producte Kronecker, de forma que el mateix píxel estigui repetit  $n$  vegades.  $\tilde{M} = M \otimes (1, 1, \dots, 1_n)^T$ .

La relació entre les matrius  $\tilde{M}$  i  $M$  és:

1. Els valors singulars no nuls de  $\tilde{M}$  són els de  $M$  multiplicats per  $\sqrt{n}$ .
2. Si  $\sigma \neq 0$  és un valor singular de  $M$ , el vector singular de sortida associat a aquest vector és igual al vector propi de  $\tilde{M}$ . És a dir,  $v_i = \tilde{v}_i$
3. Si  $\sigma \neq 0$  és un valor singular de  $M$ , amb vector de d'arribada  $u = (u_1, u_2, \dots)$ . El valor singular  $\sigma_{\tilde{M}} = \sigma_M \sqrt{n}$  té vector d'arribada:

$$\tilde{u} = \left( \frac{u_{11_1}}{\sqrt{n}}, \dots, \frac{u_{11_n}}{\sqrt{n}}, \frac{u_{12_1}}{\sqrt{n}}, \dots, \frac{u_{12_n}}{\sqrt{n}}, \dots, \frac{u_{21_1}}{\sqrt{n}}, \dots, \frac{u_{21_n}}{\sqrt{n}}, \frac{u_{22_1}}{\sqrt{n}}, \dots, \frac{u_{22_n}}{\sqrt{n}}, \dots \right).$$

Cada component de cada vector  $u_i$  d'arribada es repeteix  $n$  vegades dividit per  $\sqrt{n}$ .

### 5.1.3 Components a estudiar

Donat els resultats anteriors, es vol garantir la invariància d'escala. És per això que les components a estudiar han de ser tals que no varien tant si s'estudia  $M$  com una matriu en la qual s'ha produït un canvi d'escala  $\tilde{M}$ . Ja sigui perquè s'ha escrit la lletra més aplanada, més petita, o simplement hi ha un canvi de resolució en l'escaneig.

És per aquest motiu que aquestes components seran:

- Quocient resultants de dividir cada valor singular pel primer (el de valor més gran):  $\frac{\sigma_2}{\sigma_1}, \frac{\sigma_3}{\sigma_1}, \dots, \frac{\sigma_r}{\sigma_1}$ .
- Els vectors singulars de sortida i d'arribada, considerats com funcions de  $[0, 1] \rightarrow \mathbb{R}$ . La funció estarà dividida en  $k$  parts iguals on  $k$  és la longitud del vector i tindrà per valor constant cada component del vector. Posteriorment, es normalitza per tal que tinguin norma  $L^2 = 1$ .

Així, aquests components no variaran encara que es faci un reescalat dels eixos.

Donat que s'ha vist que a partir d'un nombre determinat de valors singulars i vectors singulars, aquests no aporten informació, els valors singulars a estudiar seran els tres primers, és a dir, els quocients:  $\frac{\sigma_2}{\sigma_1}$  i  $\frac{\sigma_3}{\sigma_1}$ .

Pel que fa als vectors singulars de sortida i d'arribada, s'estudiaran els relatius a aquests valors singulars ( $u_1, u_2, u_3$  i  $v_1, v_2, v_3$ ) de la forma expressada anteriorment com es pot observar en la figura 2.

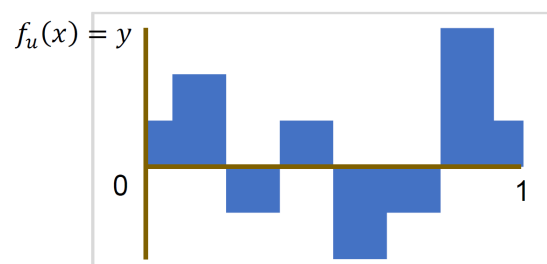


Figura 2: Funció dividida en  $k$  parts entre 0 i 1

## 6 Catàleg

La segmentació de text manuscrit conté una etapa prèvia en la qual es prepara tota la informació que ha de rebre l'algoritme per tal de poder aïllar cada caràcter individualment d'una paraula escrita de forma junta o amb lletra lligada. En la segona etapa, ja és l'algoritme pròpiament dit, és a dir, a partir d'una imatge en la qual contingui un text, extreure les lletres que hi conté.

La lletra manuscrita difereix segons la cal·ligrafia de cada persona en particular. De vegades, fins i tot als humans, ens costa entendre el que escriu qualsevol altra persona de la qual no hagem llegit anteriorment la seva cal·ligrafia personal.

De la mateixa manera, l'algoritme ha de recollir informació sobre com seran les lletres de l'individu en qüestió. A partir d'aquesta informació, posteriorment, es realitzarà la identificació de text.

És per aquest motiu que existeix un catàleg, el qual, consisteix en la descomposició SVD de l'alfabet a detectar i l'obtenció de les components a estudiar. Aquest alfabet ha de ser escrit per la persona a la qual se li ha de identificar el text i constarà de 10 mostres de cada lletra/símbol ja que la tipografia de la lletra varia.

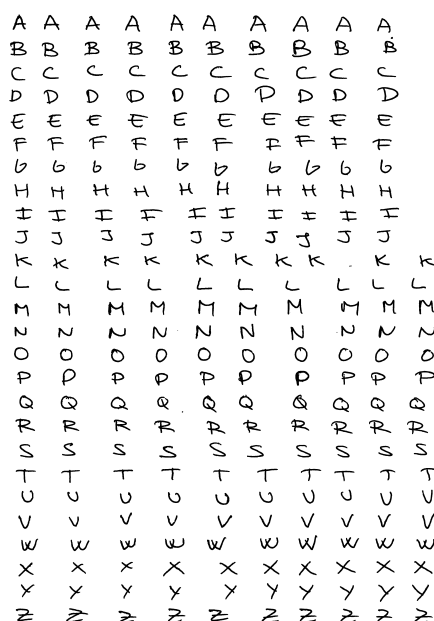


Figura 3: Exemple plantilla catàleg

Així el catàleg estarà format per lletres majúscules, minúscules, nombres i símbols que es vulguin identificar de forma aïllada. En la figura 3 es pot observar un exemple de catàleg format exclusivament per les lletres majúscules.

### 6.1 Creació del catàleg

El procés de creació del catàleg consisteix en que a partir d'una imatge i d'un arxiu amb format *.txt* amb el contingut del catàleg (quines lletres i símbols conté el catàleg), reculli les components necessàries. D'aquesta manera es guarden aquestes components que apunten a la lletra

corresponent.

En aquest cas, les lletres han d'estar separades les unes de les altres i amb separació entre línies. Com es pot observar en l'anterior figura 3.

La creació del catàleg consta de 4 etapes:

1. Correcció dels reflexos
2. Binaritzat
3. Segmentació per lletres
4. Obtenció de les components

Les quals s'expliquen en els següents subapartats.

### 6.1.1 Correcció dels reflexos

A l'escanejar una fotografia amb un escàner es produeixen reflexos que provoquen que la lletra no sigui contínua, és a dir, tot el traç que s'ha produït amb el bolígraf no estigui connectat. A banda, aquest soroll pot provocar la variació dels valors singulars i és perjudicial per a la correcta identificació de les lletres. En la següent figura 4 podem veure la lletra "f" original i la obtinguda a través del procés en aquesta etapa.

El mètode és senzill, s'aplica l'equació 7 iterativament. Sent  $p_{i,j}$ , un píxel en la posició  $i$  i  $j$  de la imatge. Llavors:

$$p_{i,j} = 0.6 \cdot p_{i,j} + 0.125 \cdot p_{i-1,j} + 0.125 \cdot p_{i,j-1} + 0.125 \cdot p_{i+1,j} + 0.125 \cdot p_{i,j+1} \quad (7)$$

És a dir cada píxel de la nova imatge conté un 60% del seu contingut més un 12,5% del contingut dels seus veïns. Obtenint el resultat següent:

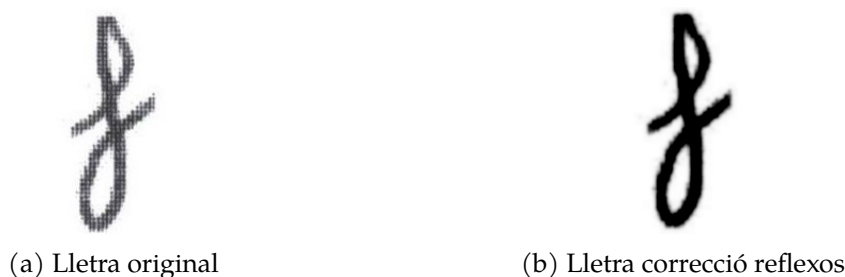


Figura 4: Procés correcció reflexos escaneig

### 6.1.2 Binaritzat

Consisteix en atribuir un valor lògic a cada píxel de la imatge, sent 0 l'absència de color (negre) i 1 la presència de color (blanc). Simplificant així la informació de la imatge i estudiant només les àrees d'interès.

Podem veure l'exemple de la lletra anterior "f" en el qual al voltant d'aquesta, ja sigui a l'exterior o en les àrees interiors queden píxels amb alts nivells de gris (és a dir quasi blancs) que donen la sensació d'imatge borrosa. Aquests píxels no aporten informació i són eliminats a través de la binarització com es veu el la figura 5.

El procés de binaritzat consisteix en establir un valor llindar a partir del qual, qualsevol píxel que el superi se li assigna el valor 1 (blanc), en cas contrari, se li assignarà el valor 0 (negre). D'aquesta forma, els píxels amb alts nivells de gris passen a ser blancs i a formar part del fons de la imatge, en canvi, els píxels amb baixos nivells de gris passen a ser negres.



Figura 5: Procés de binaritzat

### 6.1.3 Segmentació per lletres

En el cas del catàleg, les lletres estan situades de forma totalment distant unes de les altres i amb disposició horitzontal. Per a la detecció de cada lletra, en primer lloc, es detecten els salts de línia. Aconseguint els valors de posició de les files de la imatge en els quals estan contingudes totes les lletres d'una línia.

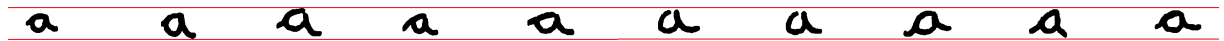


Figura 6: Delimitació de la línia

Un cop obtinguda una línia, es delimiten les lletres obtenint la posició dels dos punts que determinen el rectangle en el qual està continguda la lletra. Com es veu en la figura 7, es determina la fila i columna de la cantonada esquerra i de la mateixa manera per a la cantonada dreta inferior.



Figura 7: Delimitació de la lletra

Aquests posicions es guarden en una matriu, per tal utilitzar-les com a variable d'entrada per a la següent etapa.

#### 6.1.4 Obtenció de les components

Per realitzar la comparació entre els caràcters del catàleg i la del text es realitza la descomposició en valors singulars a partir de la posició de cada lletra obtinguda en l'etapa anterior. Com ja s'ha explicat en l'apartat 5.1.3, es guarden els valors corresponents a:

- La relació del segon i tercer valor singular amb el primer:  $\frac{\sigma_2}{\sigma_1}, \frac{\sigma_3}{\sigma_1}$ .
- Els tres primers vectors singulars d'arribada:  $u_1, u_2, u_3$ .
- Els tres primers vectors singulars de sortida:  $v_1, v_2, v_3$ .

Una altra component que es desa és la relació màxima i mínima entre l'alçada i amplada de les lletres. Aquesta no és necessària per al reconeixement d'un caràcter, però com es veure posteriorment en l'apartat 8.3 s'utilitza per a buscar lletres dins d'una paraula.

A través d'aquestes dades i sabent les lletres que estan incloses al catàleg ja es disposa la informació necessària que representa cada lletra.

## 7 Reconeixement de lletra manuscrita

Quan ja es disposa del catàleg, els passos a seguir per a la identificació de text són els següents:

1. Correcció dels reflexos
2. Binaritzat
3. Segmentació per paraules
4. Segmentació per lletres
  - (a) Cerca iterativa de lletra candidata.
  - (b) Comparació amb el catàleg de la candidata.
  - (c) Assignació candidata.
5. Transcripció a text digital

Les dues primeres etapes comparteixen el funcionament amb la creació del catàleg, vist en els apartats [6.1.1](#) i [6.1.2](#).

En el cas de la segmentació per paraules, el funcionament també és compartit amb l'apartat [6.1.3](#) però en aquest cas enlloc de segmentar lletres ho fa amb paraules. Això es degut a que es separen blocs de píxels escrits de forma contínua, en el cas del catàleg al estar les lletres separades, segmenta les lletres. En aquest cas tenim un text escrit amb lletra lligada, per tant, el que s'escriu de forma contínua són les paraules.

### 7.1 Segmentació de lletres

En aquest punt s'ha obtingut les coordenades que delimiten la paraula (com en el cas del catàleg però enlloc de delimitar una lletra en aquest cas una paraula, veure figura [7](#)) ara toca a identificar cada lletra que pertany en aquesta paraula.

Com es veu en la següent figura [8](#), ja es disposaria de la paraula aïllada [8a](#) i s'ha de segmentar cada lletra i identificar correctament quina és cadascuna ([8b](#),[8c](#),[8d](#),[8e](#),[8f](#),[8g](#)).

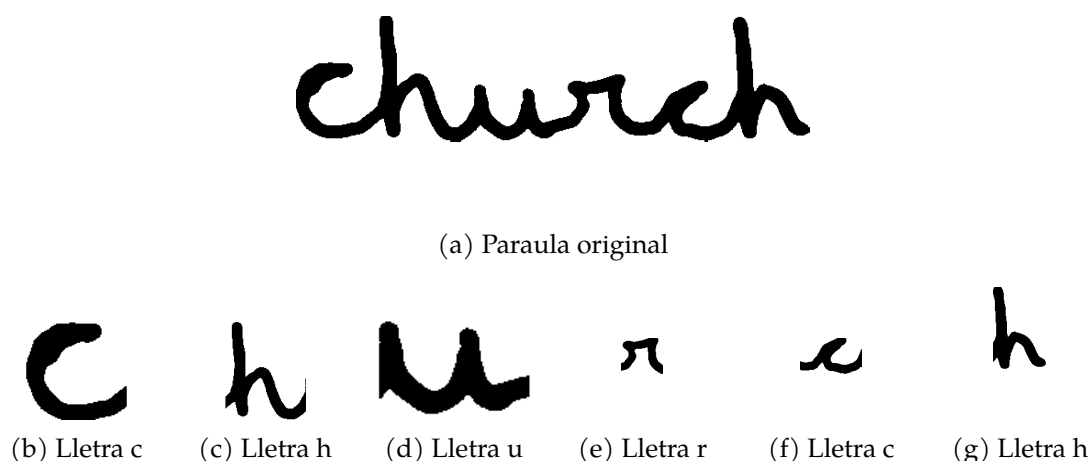


Figura 8: Segmentació de lletres dins d'una paraula

### 7.1.1 Cerca iterativa de lletra candidata

Per tal de cercar una lletra dins de la paraula, el procés consisteix dividir-la. Cada divisió és un candidat a possible lletra. Aquest trossejat s'anomenarà malla.

Per determinar quants de trossos cal dividir la paraula, es realitza a partir d'un cert nombre de píxels determinat, ja que dividir paraules de diferents longituds amb el mateix número d'interval·ls no seria idoni. Això, comportaria que les paraules curtes estiguessin massa mallades, tenint un alt nombre de candidats i més temps de càlcul. També, seria dolent en el cas contrari, ja que seria provable que cap trossejat coincidís exactament amb la lletra.

A continuació es pot veure una paraula amb un excés de malla, i amb un manca de mallat:

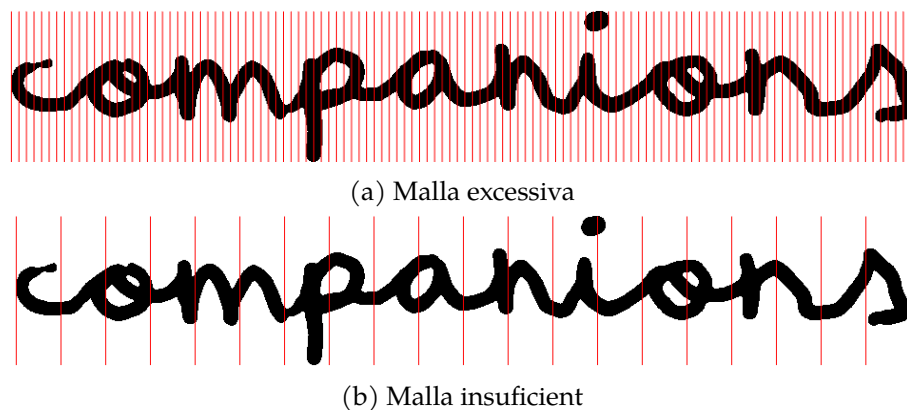


Figura 9: Diferents tipus de mallat

Un cop definida la malla, es delimita la possible candidata de forma que no quedin píxels negres per sobre ni per sota. Com es veu en la següent figura 10, es mostren les 9 primeres candidates, aquestes s'analitzarien fins al final de paraula.



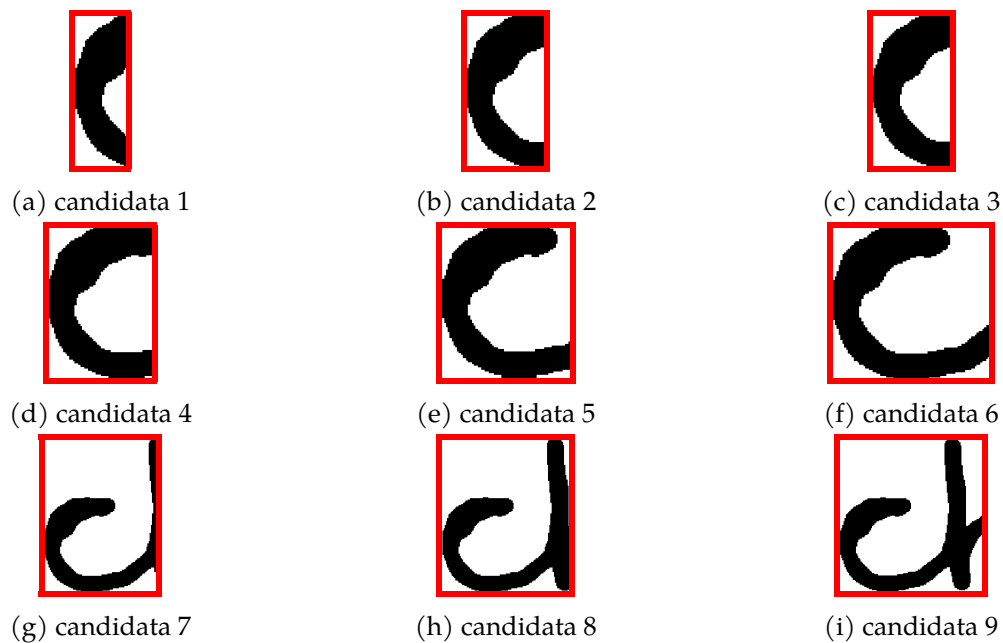


Figura 10: Candidates a lletra

Aquestes candidates passen a la següent etapa, la qual s'explica en el següent subapartat, en les quals es determina la millor candidata i a quina lletra correspon. Tal com es veu en la figura 11a, la paraula ha estat mallada i s'ha anat analitzant candidata per candidata, com es veu en la figura 10. Per finalment, arribar a la conclusió que la candidata idònia és la "c" (11b).

Una vegada determinada la candidata ideal de la primera lletra, es procedeix a buscar la candidata ideal de la segona. Com es veu en la següent figura 11c, s'inicia la cerca a partir de la línia delimitada en blau, la qual coincideix amb el final de la lletra anterior trobada. Així successivament fins a trobar totes les lletres corresponents a la paraula.

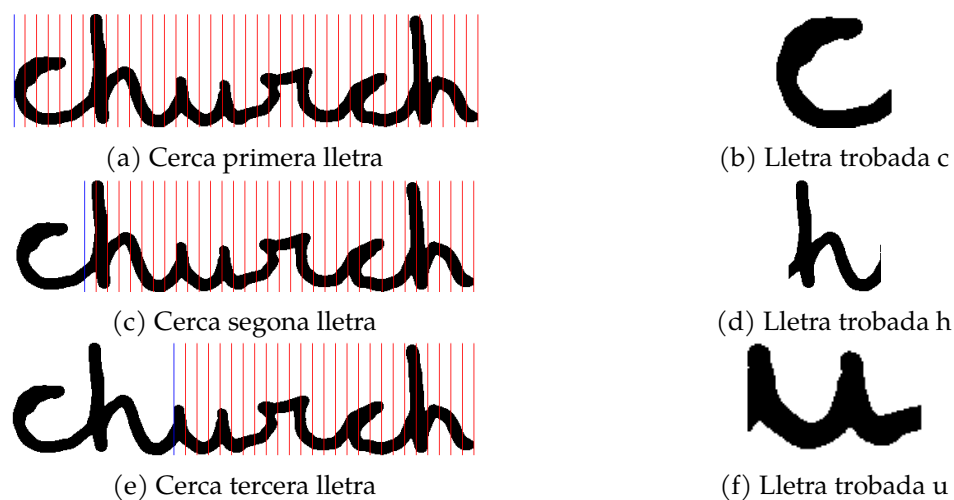


Figura 11: Procés de búsqueda de candidates

### 7.1.2 Comparació de la candidata amb el catàleg

Totes les candidates passen per aquest procés en el qual es comparen les components a estudiar de cada candidata amb les del catàleg, obtenint una distància de la candidata amb cada lletra del catàleg.

Com les components a estudiar són els invariants obtinguts dels valors singulars i dels vectors singulars, s'ha de realitzar una ponderació d'aquests.

Les ponderacions utilitzades, són les següents:

- Ponderació valors singulars:  $taxa^{vs} = [1, 0.4]$
- Ponderació vectors singulars de sortida:  $taxa^u = [0.35, 0.3, 0.15]$
- Ponderació vectors singulars d'arribada:  $taxa^v = [0.35, 0.3, 0.15]$

Per determinar-les ens hem basat en l'optimització del nombre d'encerts de l'algoritme d'identificació lletra a lletra de l'anterior projecte, tenint com a variables aquests valors de ponderació.

L'ordre decreixent es deu a que com s'ha vist en 5, la informació aportada pels primers valors i vectors singulars és major.

#### 7.1.2.1 Comparació valors singulars

El procés és senzill, es calcula la diferència entre els invariants del valors singulars, és a dir, entre els invariants de la lletra del catàleg que s'està comparant i la lletra candidata. Obtenint un vector de dos components, a les quals es realitza la ponderació anteriorment mencionada.

Finalment es calcula la norma, obtenint un únic valor com a distància.

#### 7.1.2.2 Comparació valors singulars

S'ha de tenir en compte que la longitud dels vectors singulars pot no ser igual. Aquesta variarà per a cada lletra segons la dimensió de la seva imatge. Per aquest motiu s'adapten els dos vectors per a que tinguin la mateixa longitud.

Donat el vector  $v$  (vector singular de la lletra candidata) de longitud  $n$ ; i  $c$  (vector singular de la lletra del catàleg) de longitud  $m$ . Cada valor del vector  $v$  quedarà repetit  $m$  vegades. També, per al vector  $c$  cada valor quedarà repetit  $n$  vegades, obtenint dos vectors de longitud  $n \cdot m$ .

A més, tenint en compte les propietats d'invariància vistes a 5.1.3, cada vector queda multiplicat per l'arrel de la longitud. Quedant de la següent manera:

$$v = (v_1, v_2, \dots, v_n);$$

$$c = (c_1, c_2, \dots, c_m)$$

$$v_{adaptat} = \sqrt{m} \cdot (v_{1,1}, v_{1,2}, \dots, v_{1,m}, v_{2,1}, v_{2,2}, \dots, v_{2,m}, \dots, v_{n,1}, v_{n,2}, \dots, v_{n,m});$$

$$c = \sqrt{n} \cdot (c_{1,1}, c_{1,2}, \dots, c_{1,n}, c_{2,1}, c_{2,2}, \dots, c_{2,n}, \dots, c_{m,1}, c_{m,2}, \dots, c_{m,n});$$

Un cop igualat aquests vectors amb longitud i tenint en compte les propietats d'invariància, es procedeix a calcular la distància  $d$  entre aquests dos vectors:

$$d = \frac{1}{\sqrt{n \cdot m}} \min \left( \|v_{adaptat} - c_{adaptat}\|_2; \|v_{adaptat} + c_{adaptat}\|_2 \right) \quad (8)$$

Ja explicat com es calcula aquesta distància per a un vector singular, el procediment és idèntic tant si es tracta d'un vector singular d'entrada com de sortida.

Finalment, es pondera aquesta distància de la següent manera:

$$d_{vectors\ singulars} = \sum_{i=1}^3 \sqrt{(taxa_i^u \cdot d_i^u)^2 + (taxa_i^v \cdot d_i^v)^2} \quad (9)$$

### 7.1.3 Elecció de la millor candidata

Finalment es sumen les distàncies dels valors singulars i dels vectors singulars, obtenint la distància total. Aquesta distància global serveix com a indicador de quina lletra és la més semblant del catàleg. Així, cada candidata es compara amb totes les lletres del catàleg, per quedar-se amb la lletra que minimitza la distància.

D'aquesta manera per a cada una proposta de lletra (la més semblant del catàleg, és a dir, la que minimitza la distància), quedant-nos un altre cop amb la candidata de menor distància a la lletra proposta del catàleg.

## 7.2 Transcripció a text digital

El programa identifica blocs de lletres escrites sense separació horitzontal. Per cada columna de píxels d'una línia, analitza si conté algun píxel negre. En cas que el següent píxel negre es trobi en una nova columna, es considerarà de la paraula següent alhora de fer la segmentació de lletres. Tot i així, pot ser que formin part de la mateixa paraula. Com per exemple en la següent figura 12.

El que es vol dir és que el programa no entén de molta separació o poca, és per això que alhora de transcriure les lletres identificades amb els espais corresponents s'ha de tenir en compte aquest fet.



Figura 12: Paraula amb separació vertical

Així, s'ha estudiat la distribució probabilística dels espais. Aquests segueixen distribució una normal. S'ha vist que els valors observats d'espais amb distància menor a 50 píxels són els valors que corresponen a lletres que formen part de la mateixa paraula.

És per aquest motiu que es considerarà com a lletres consecutives que formen part de la mateixa paraula totes aquelles que siguin menors a la mitjà de la distància dels espais menys 3 vegades la desviació tipus. On abans s'han exclòs tots aquells valors menors 50 píxels.

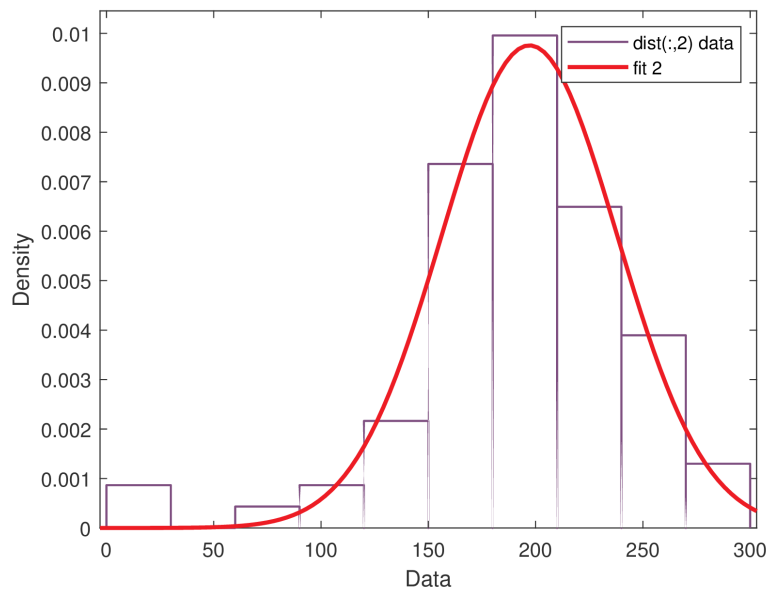


Figura 13: Distribució espais excloent valors menors de 50 píxels

## 8 Millora de la taxa d'encert

En aquest apartat s'explicarà la manera d'avaluar el tan per cent d'encerts. A més, s'explicaran les millores introduïdes que han fet possible millorar l'algorisme inicial ja que la taxa d'encert menor al 20%.

### 8.1 Avaluació de la taxa d'encert

Avaluar la quantitat de lletra reconeguda a un text no és senzill, ja que no es pot comparar lletra a lletra. Això és degut a que donada una paraula de longitud de sis lletres pot ser que se n'identifiquin vuit a causa dels errors. Per exemple, si la paraula a reconèixer és "act" i el programa identifica la "ct" com una "d". Quants errors s'han realitzat? Un al considerar "ct" com "d" o dos al no identificar correctament la "c" i la "t"?

Com s'observa, s'ha de definir un mètode per poder mesurar com de ben reconeguda és una paraula. Un dels mètodes més utilitzats per mesurar la similitud entre paraules i textos és la distància de Levenshtein [7].

La distància de Levenshtein consisteix en comptabilitzar el número de insercions, canvis o supressions de lletres necessàries per tal de convertir una lletra amb una altra. A continuació es mostren alguns exemples per poder entendre com calcula la distància:

- **Espasa - Esposa:** Distància 1 (canvi de la "a" per la "o")
- **Casa - Casat:** Distància 1 (supressió de la "t")
- **Company - Campanya:** Distància 2 (canvi de la "o" per la "a" i addició de la última a).
- **Levenshtein - leuennshtei:** Distància 4 (canvi de la "L" majúscula per la "l" minúscula, canvi de la "u" per la "v", supressió de una "n" i addició de la última n).

Així, per tal d'avaluar la taxa d'encerts el càlcul és realitzarà de la següent manera:

$$Taxa\ encerts = \frac{Llargada\ text\ original - lev(text\ original; text\ reconegut)}{Llargada\ text\ original} \quad (10)$$

On *lev* és la funció que calcula la distància de Levenshtein.

Per tant en el cas anterior, on es comparava "act" amb "ad" la taxa d'encert seria del 33%:

$$Taxa\ encerts = \frac{3 - lev("act"; "ad")}{3} = \frac{3 - 2}{3} = \frac{1}{3}$$

### 8.2 Creació d'un catàleg representatiu de lletra lligada

Realitzant el mètode explicat anteriorment la taxa d'encert no arriba al 20%. El programa segmenta algunes lletres correctament però no és capaç de d'encertar quina lletra és l'escripta. Sembla que no reconeix les lletres. O solament un petit % d'aquestes.

Per tant, es va decidir comprovar si el plantejament que es duia a terme era l'adequat. Per fer-ho es va decidir realitzar un catàleg de lletra lligada escrita per ordenador i analitzar lletra escrita

amb aquest catàleg (concretament la lletra *Segoe Script*). Al realitzar aquesta prova es va veure com la taxa d'encert millorava fins a un 68%. S'esperava una taxa de reconeixement major ja que es tractava de lletra lligada escrita per un ordinador amb el qual no hi ha variacions.

Tot i això, aquest test va ser de vital importància per poder tirar endavant el projecte, ja que va confirmar dues coses:

- El plantejament utilitzat era adequat ja que la taxa de reconeixement va augmentar considerablement al no haver-hi variacions d'escriptura.
- Hi havia una percentatge de text que no era capaç de reconèixer.

La clau estava en aquest últim punt. Es va poder observar que les lletres del catàleg no eren exactament iguals a les lletres que s'havien d'identificar, ja que al ser lletra lligada part d'una lletra forma part de la lletra següent. Com es pot veure en la part dreta de la següent figura 14, la cua de la a forma part de la p, la cua de la p forma part de la o...

Les lletres que hi ha al catàleg són les lletres separades (les de l'esquerra de la figura), és a dir, sense tenir en compte que anteriorment hi ha una altra lletra.



Figura 14: Diferència lletra lligada amb lletra separada

Observat aquest problema queda clar que disposar d'un catàleg de lletra lligada és de vital importància i al mateix temps més complicat del que sembla.

En el cas de la lletra lligada s'ha d'anar amb molt de compte ja que les lletres no només varien pel propi fet d'estar escrites de forma manuscrita. A més a més, es té l'afegit de la variabilitat deguda a les condicions de contorn de la lletra. Depenent per quina lletra ve precedida la paraula, la cal·ligrafia és diferent.

Aquest fet no s'havia apreciat prèviament, així que s'ha de procedir a realitzar un catàleg que ho tingui en compte. Així, sorgeixen dues vies possibles per tal de realitzar el catàleg:

- Eliminar les cues de les lletres per elaborar un catàleg només tenint en compte la part central de la lletra.
- Determinar quines són les possibles condicions de contorn que provoquen que la lletra variï i elaborar un catàleg tenint en compte aquestes possibles condicions de contorn.

De les dues opcions, s'ha decidit a desenvolupar la segona. El següent pas, és doncs identificar quines són les condicions de contorn que provoquen que una lletra s'escrigui de diferent forma.

S'han identificat 3 possibles condicions de contorn que provoquen que la lletra canviï. Aquestes condicions s'han parametritzat com:

- Inici de paraula.
- Lletra precedida per lletra amb cua final superior. Precedida per lletres tipus “b”, “v”, etc.
- Lletra precedida per lletra amb cua final inferior, precedida per lletres tipus “j”, “a”, etc.

Aquestes condicions es poden veure en l'exemple següent:

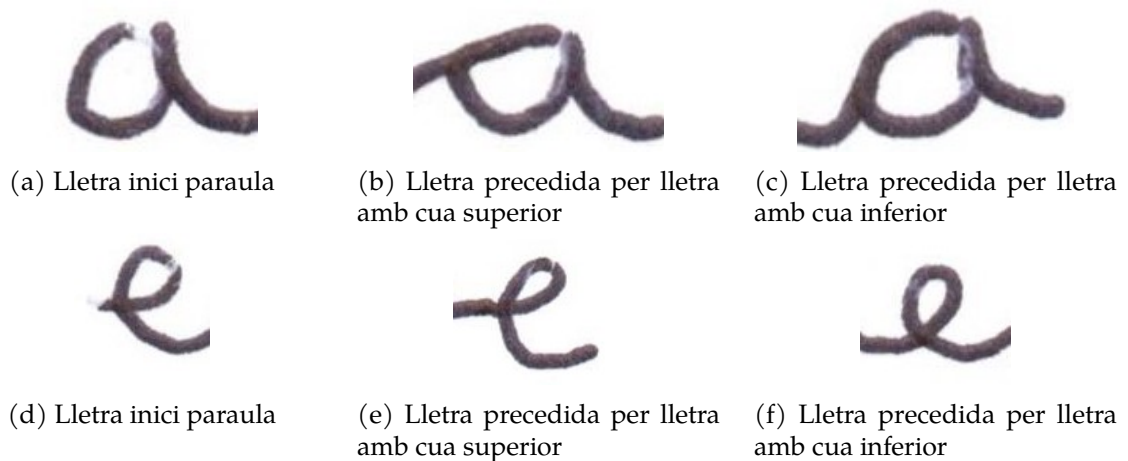


Figura 15: Lletres amb diferents condicions de contorn

Amb aquesta premissa s'ha realitzat un catàleg repetint com s'ha dit anteriorment cada lletra 10 cops. Intentant tenir almenys dos mostres de cada condició de contorn.

Per poder realitzar aquest catàleg s'ha hagut de retallar manualment cada lletra. Extraient-les de dintre de paraules senceres per poder subministrar al programa una informació més rellevant. Un cop retallades amb qualsevol programari d'edició d'imatges que permeti retallar, s'insertien amb la mateixa mida original a un fitxer per a posteriori imprimir-ho.

### 8.2.1 Eliminació de valors atípics

Per tal de poder millorar la taxa de reconeixement es va procedir a identificar si hi havia lletres del catàleg que induïen clarament a error (*outliers*).

Es van identificar dos *outliers*, les quals es va decidir eliminar del catàleg i substituir-les per dos altres. Com es pot veure, la primera lletra “c” (veure figura 16c), realment és una c poc representativa que és lògic que doni error, a més, el programa ho confonia usualment amb una u. La segona “c” (figura 16f) desencadenava molts errors posteriors ja que usualment era confosa amb “a” o “o” partint-les per la meitat (figura 16e).

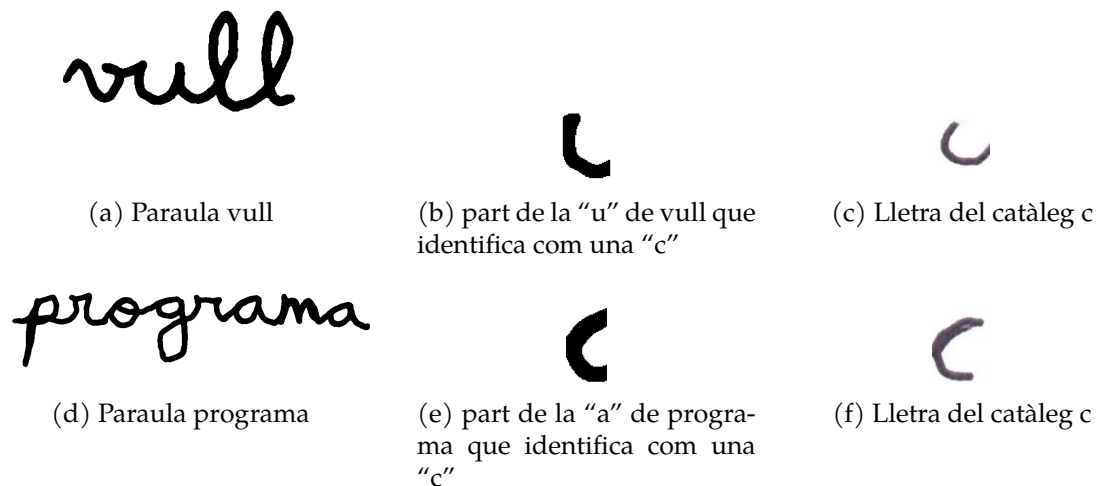


Figura 16: Lletres que provoquen errors

Altres fets que es varen observar d'aquest detallat anàlisi van ser els següents:

- La taxa d'encert de lletres majúscules era elevada (al voltant del 80%)
- Tot i que no s'identifiquen més lletres del catàleg com a valor atípics, s'observa que lletres poc freqüents com la "x" o la "w" normalment compten les seues aparicions com a errades. Sense tenir cap lletra del catàleg com a responsable d'aquests errors.

### 8.2.2 Catàleg segons la freqüència d'aparició de la lletra

És habitual al tractar en elements físics o algorismes, tenir en compte que l'aparell o el software que realitzaràs està enfocat a una població, de la qual tenir una mostra elevada d'informació estadística és important per poder adaptar-lo de forma adequada.

Amb aquesta idea es dissenya qualsevol aparell ergonòmic, com per exemple, el teclat de l'ordinador, el qual la disposició de les lletres i símbols es estudiada per poder oferir una ràpida escriptura.

Una idea que va sorgir va ser realitzar un catàleg representatiu amb la freqüència en la qual les lletres apareixen en els textos. És a dir, enlloc de fer 10 repeticions de cada lletra, fer-ne més de les vocals i de les lletres més freqüents i menys de les lletres menys freqüents com la "z".

Com aquest treball no pretén identificar accents, es va basar en la freqüència de les lletres de l'anglès. Tot i així, la correlació amb l'espanyol és del 95% [8]. Les freqüències mostrades a la taula 1 també han estat obtingudes de [8], en el qual l'autor analitza diversos textos fins arribar a un total de 806 milions de lletres per a la llengua anglesa.

D'aquesta manera es podria reduir la taxa d'errors deguda a l'aparició de lletres poc freqüents i millorar la taxa d'encert de les lletres més freqüents.

Es van idear tres catàlegs:



Catàleg 0:

El catàleg 0 era el catàleg plantejat inicialment en el qual cada lletra té 4 mostres diferents.

Catàleg 1:

El contingut del catàleg és directament proporcional a la freqüència d'aparició de la lletra, amb un mínim d'una lletra. S'ha intentat conservar el nombre de lletres per catàleg del catàleg original (d'aquí el 260). Per tant, el nombre de mostres per catàleg s'ha determinat de la següent manera:

$$Mostres lletra_i = \begin{cases} 1 & \text{si } \text{arrodoneix}(freq_i \cdot 260) < 1 \\ \text{arrodoneix}(freq_i \cdot 260) & \text{si } \text{arrodoneix}(freq_i \cdot 260) > 1 \end{cases}$$

Catàleg 2:

En aquest cas, s'ha volgut observar si el fet de tenir moltes mostres d'una lletra és tenir sobreinformació o realment és informació útil, a més, de pal·liar el dèficit de lletres poc freqüents. S'ha establert doncs un llindar màxim i un llindar mínim de lletres per catàleg.

$$Mostres lletra_i = \begin{cases} 4 & \text{si } \text{arrodoneix}(freq_i \cdot 260) < 4 \\ \text{arrodoneix}(freq_i \cdot 260) & \text{si } 4 < \text{arrodoneix}(freq_i \cdot 260) < 15 \\ 15 & \text{si } 15 < \text{arrodoneix}(freq_i \cdot 260) \end{cases}$$

El resum de mostres de cada lletra per a cadascun dels catàlegs es pot veure a la següent taula [1](#).

Com s'ha vist en [8.2.1](#), la taxa d'encert de les lletres majúscules és elevada, per aquest motiu es decideix provar a reduir fins a 4 mostres per cada lletra majúscula. Així doncs, afegint un total de 104 elements més a cada catàleg ( $26 \text{ lletres} \cdot 4 \text{ mostres/lletra}$ ).

| Lletra | Freqüència (%) | Mostres catàleg 1 | Mostres catàleg 2 | Mostres catàleg 0 |
|--------|----------------|-------------------|-------------------|-------------------|
| e      | 12,15          | 32                | 14                | 10                |
| a      | 8,67           | 23                | 14                | 10                |
| t      | 8,60           | 22                | 14                | 10                |
| i      | 7,53           | 20                | 14                | 10                |
| o      | 7,38           | 19                | 14                | 10                |
| n      | 7,34           | 19                | 14                | 10                |
| s      | 6,63           | 17                | 14                | 10                |
| r      | 6,62           | 17                | 14                | 10                |
| h      | 4,75           | 12                | 12                | 10                |
| l      | 4,24           | 11                | 11                | 10                |
| d      | 3,90           | 10                | 10                | 10                |
| c      | 3,42           | 9                 | 9                 | 10                |
| u      | 2,69           | 7                 | 7                 | 10                |
| m      | 2,65           | 7                 | 7                 | 10                |
| f      | 2,30           | 6                 | 6                 | 10                |
| p      | 2,12           | 6                 | 6                 | 10                |
| g      | 1,95           | 5                 | 5                 | 10                |
| w      | 1,67           | 4                 | 4                 | 10                |
| y      | 1,55           | 4                 | 4                 | 10                |
| b      | 1,49           | 4                 | 4                 | 10                |
| v      | 1,06           | 3                 | 4                 | 10                |
| k      | 0,60           | 2                 | 4                 | 10                |
| x      | 0,20           | 1                 | 4                 | 10                |
| j      | 0,19           | 1                 | 4                 | 10                |
| z      | 0,13           | 1                 | 4                 | 10                |
| q      | 0,11           | 1                 | 4                 | 10                |
| Total  | 99,95          | 263               | 221               | 260               |

Taula 1: Freqüència d'aparició de lletres [8] i mostres per catàleg

### 8.3 Reducció de lletres candidates

El fet de mallar tota la paraula i analitzar tots els petits trossos fins al final comporta un cos computacional molt alt. A més, pot arribar a provocar errors innecessaris que de produir-se tenen un impacte important ja que, per exemple, prendre 3 lletres com a una sola provoca que augmenti la distància de Levenshtein. No només augmenta 2 punts, el més provable és que augmenti a 3 ja que s'hauran de inserir dos lletres i la lletra reconeguda segurament no serà cap de les 3.

D'avant d'aquest fet la idea que ha sorgit és tenir en compte la relació  $\frac{\text{amplada}}{\text{altura}}$ .

Si es té la informació de quina és la relació mínima amplada-altura d'un catàleg es té una informació molt valiosa. Les lletres que acostumen a tenir aquesta relació baixa són les lletres que són menys amples i més altes com la "f", la "l" o la "j". Si es multiplica l'alçada de la lletra per aquesta relació mínima és té des de quin punt es pot començar a buscar una lletra.

D'altra banda, en el cas contrari, les lletres que tenen una relació molt alta, són amples i poc

altes, com la “m”. Si multipliquem l'alçada de la “m” per aquesta relació màxima obtindrem una cota màxima de fins on pot haver-hi una lletra.

El problema és que quan es comença a analitzar un text no se sap si s'escriuran les lletres com en el catàleg o, en cas contrari, més grans. Tampoc se sap si la resolució de la imatge serà la mateixa. No es té doncs l'alçada d'aquestes lletres, per tant, s'ha d'aproximar doncs aquesta mesura estadísticament.

A partir de les dades que es van enregistrant de les lletres que es detecten, s'estimen aquests valors. Per determinar-los s'ha tingut en compte la seva distribució.

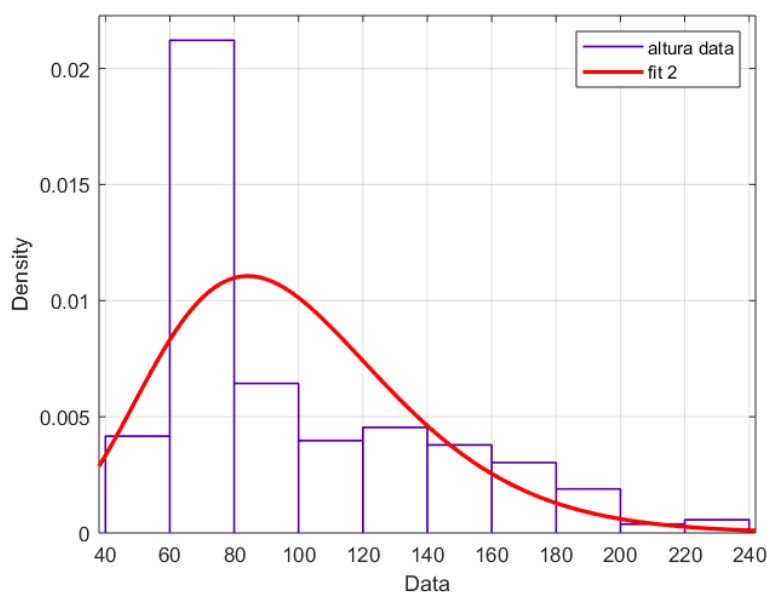


Figura 17: Distribució altura lletres

Donat que una lletra amb una relació màxima és una lletra amb poca alçada, s'aproxima aquesta lletra com:

$$\text{altura lletra relacio minima} \approx \mu - \frac{\sigma}{2}$$

On  $\mu$  és la mitja de l'alçada de les lletres i  $\sigma$  la desviació tipus.

De manera semblant es realitza per a la lletra amb relació mínima, és a dir una lletra tipus “f”.

$$\text{altura lletra relacio maxima} \approx \mu + 2\sigma$$

Aquestes aproximacions, es comencen a aplicar un cop es tenen suficients dades. Mentre no es tenen dades s'utilitza l'alçada de la paraula com a estimació de l'altura de la lletra amb relació màxima, la qual sol ser un bon indicador. En el cas de la relació mínima també s'utilitza l'amplada de la paraula. Solament, provocarà que al inici realitzi un mallat més extens i conforme tingui suficients dades s'adapti.

A la següent figura 18, es pot veure la substancial millora que s'aconsegueix amb aquest canvi.

S'analitzen moltes menys candidates, el programa va més ràpidament i s'aconsegueixen reduir les probabilitats d'error.

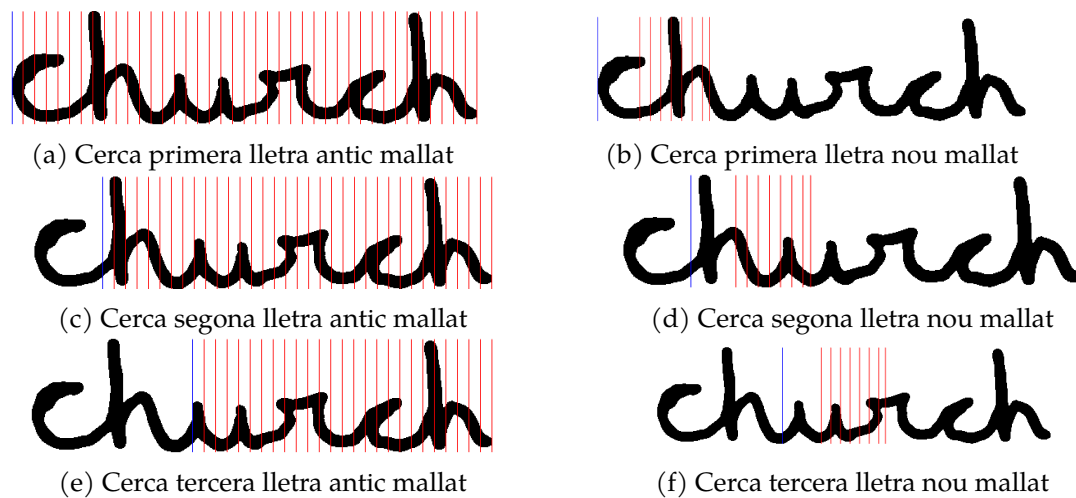


Figura 18: Nou procés de búsqueda de candidates

#### 8.4 Elecció del millor catàleg i resultats obtinguts

Per tal d'analitzar quin era el millor catàleg es van comparar els tres catàlegs explicats anteriorment amb el mateix text. La taxa d'encert obtinguda per a cadascun dels textos va ser la següent:

- Catàleg 0 (10 mostres de cada lletra): Taxa d'encert del 66%.
- Catàleg 1 (proporcional): Taxa d'encert del 64%.
- Catàleg 2 (proporcional, mínim 4 mostres, màxim 14): Taxa d'encert del 68%.

La taxa d'encert no varia molt entre ells ja que en la seva majoria contenen les mateixes lletres, tot i així, es pot observar com el catàleg amb menor número de lletres escrites, és el que funciona millor. A més, s'observa com tenir molta informació de les lletres més comuns no condueix a major taxa d'èxit, sinó, que amb 14 mostres són més que suficients.

A continuació, es mostra el text reconegut de la figura 19:

*th tol uecs surprixed at his thougNk Maybe the church wdth the sycanore lrowing fnon withcn had toen  
hauded It caused him to hsve the same dwrn ki a second time and it wcs causkg hcm to feel anyer towcnls  
hss fodhful companions He deenk a bd fnom the wine that remained feom the night bctne and he gdheed  
hes tadcd chesee to hcs bodg Hh kneu that a feu homs hos nau wth tdr sun at ds zenitdc vexed ngmFs go  
fox pidk asaltne bte*

the boy was surprised at his thoughts Maybe  
 the church with the sycamore growing from  
 within had been haunted It caused him to  
 have the same dream for a second time and it  
 was causing him to feel anger towards his  
 faithful companions He drank a bit from the  
 wine that remained from the night before  
 and he gathered his jacket closer to his  
 body He knew that a few hours from now  
 with the sun at its zenith

vexed nympts go for quick waltz job

Figura 19: Text a reconèixer

Com es pot apreciar en el text reconegut es produeixen dos tipus d'errades diferents.

En primer lloc, errades degudes al reconeixement erroni d'una lletra, és a dir, segmenta bé la lletra (delimita bé la posició d'una lletra) però el programa "diu" que és una altra (exemple figura 21g, on el programa identifica una "g" com una "l").

En segon lloc, errades de segmentació, és a dir, no es delimita correctament la lletra. Si es produeix aquest tipus d'error, la lletra és probable que s'identifiqui malament. I no només això, sinó que al delimitar-la incorrectament provoca un encadenament d'errades fins que el programa no s'alinea o s'acaba la paraula. Com es pot veure en la següent figura, en la qual identifica una "w" com una 'u' i provoca que la distància de Levenshtein augmenti 3 punts. Dos deguts a la lletra u i dos errors que ha desencadenat posteriorment (veure figures 20b 20b)



Figura 20: Procés d'encament d'errors de la paraula was

És per aquest motiu que aquesta taxa d'encert aconseguida és considerada molt positivament ja que donat un error del segon tipus en desencadena més.

Un cop analitzat varis texts es s'observa com les errades més freqüents són:

- "w" → u
- lletra+"t" → "d"
- "b" → "t"
- "g" ↔ "y"

A continuació es mostren algunes errades del text mostrat a la figura 19:

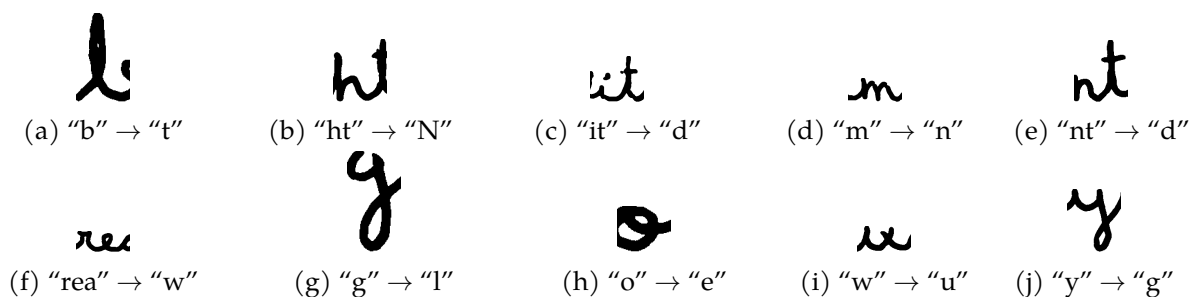


Figura 21: Alguns errors ordenats en l'ordre en el qual apareixen en el text

Com es pot veure en la figura 21h, en aquest cas confon la "o" per la "e" ja que una simetria horitzontal de la lletra semblaria analitzada sembla una lletra "e", i per tant comparteixen valors singulars.

## 9 Implementació del codi

L'implementació del codi està dividida en dues parts: per la formació del catàleg i pel reconeixement.

### 9.1 Codi per a la creació del catàleg

Per la creació del catàleg és necessari crear una carpeta dins del directori *Plantilles*, dins d'aquesta s'enganxarà una imatge del text escanejat i un fitxer de text. L'únic requisit és que els tres arxius tinguin el mateix nom.

El següent pas, és executar l'script *correr\_fitxer\_crear\_cataleg*, el qual executarà les funcions necessàries per poder crear el catàleg. L'únic que s'ha indicar és el nom de la carpeta (el mateix que el de la imatge i l'arxiu text), el format o extensió de la imatge i el nom del catàleg que es crearà.

Les funcions principals per crear el catàleg són:

- ***aplicar\_flux***: A partir de la imatge de partida obtinguda directament de l'escàner, aquesta funció aplica la primera i segona etapes explicades en [6.1.1](#) i [6.1.2](#).
- ***segmentacio***: Donat un full amb les lletres escrites de forma separada identifica la posició de cadascuna. És a dir, s'obté una matriu de 4 columnes i amb tantes files com lletres conté el catàleg. On s'emmagatzemen les 4 coordenades del rectangle que conté cada lletra.

Correspon a la tercera etapa explicada en [6.1.3](#).

- ***signatures***: Donades les posicions de totes les lletres, es calculen les components a estudiar. Correspon a la quarta etapa (veure [6.1.4](#)).

### 9.2 Codi per al reconeixement

El procés de reconeixement de lletra manuscrita lligada està compost de diverses funcions. Però el *core* del programa té lloc a la funció *HWR* la qual s'encarrega de realitzar tot el procés de reconeixement.

Per reconèixer un text, només s'ha de guardar una carpeta amb la imatge i el text original dins del directori *Imatges*. La imatge, el text i la carpeta han de tenir el mateix nom. Posteriorment, executar l'script *corre\_fitxer\_reconeixement*.

A banda d'aquesta funció principal, existeixen altres funcions importants pel funcionament de l'algoritme. Aquestes són:

- ***aplicar\_flux***: mateixa funció que per crear el catàleg. Corregeix els reflexos provocats per l'escaneig i binaritza la imatge.
- ***HWR\_ind***: Calcula la distància de la candidata a totes les lletres del catàleg.
- ***separa\_paraula***: Transforma el text reconegut i el converteix a text digital. Escrivint els espais corresponents i unint lletres en cas que, tot i no estar lligades, estiguin molt properes

l'una a l'altra.

*aplicar\_flux*, *HWR\_ind* i *separa\_paraula*.

També existeixen altres funcions secundàries necessàries per al funcionament del programa:

- ***flux\_tinta\_negra***: subrutina de la funció *aplicar\_flux*, la qual aplica la equació 7 iterativament a la imatge.
- ***lev***: Calcula la distància de Levenshtein donats dos *strings*.
- ***fusiona\_catalegs***: Aquesta funció uneix dos catàlegs en un. És útil si per a la creació d'un catàleg no caben totes les lletres dins una pàgina. O si es té un catàleg de lletres majúscules i minúscules unir-los per crear-ne un únic.
- ***taxa\_encerts***: Calcula la taxa de reconeixement del text.
- ***intervals\_zero***: Subdivideix el vector donat en subinterval·ls de píxels consecutius blancs o negres i retorna la posició d'aquests interval·ls. Aquesta funció es crida per saber on comencen i acaben les línies de la imatge.



## 10 Pressupost

El cost associat a aquest projecte es pot dividir, en diversos conceptes:

- Cost de personal: 340 hores d'enginyer en pràctiques.

$$Cost\ de\ personal = 340\ h \cdot 20\ €/h = 6800\ €$$

- Cost de llicències: En aquest cas s'ha fet servir la llicència d'estudiant. En cas d'utilitzar el programari de pagament el seu cost és de 800€ l'any.

$$Cost\ Llicencia = Valor \cdot \frac{mesos\ us}{mesos\ utilitat} = 800€ \cdot \frac{6}{12} = 400€$$

- Cost de material: El material utilitzat ha estat una impressora, un ordinador i material d'oficina.

$$Costmaterial = Ordenador + impressora + material\ oficina = 1000 + 100 + 10€ = 1110€$$

- Cost energètic

$$Cost\ energetic = P \cdot t \cdot preu = 0.08\ Kw/h \cdot 340h \cdot 0,11\ €/Kwh = 3\ €$$

- Cost total associat

$$Cost\ total = 6800 + 400 + 1110 + 3 = 8313€$$



## 11 Impacte ambiental

Aquest treball ha estat desenvolupat a través de programari informàtic, és per aquest motiu que l'impacte ambiental que hagi pogut causar és baix.

El material necessari per realitzar aquest treball ha un ordinador i una impressora.

En els aparells informàtics, l'impacte que realitzen es localitza en tres fronts:

- L'impacte ambiental de la producció de l'aparell
- El consum per fer funcionar l'aparell
- L'impacte ambiental al llençar l'ordinador a l'abocador

El problema és que els ordenadors és un producte amb un gran volum de producció a nivell mundial, fet que provoca que aquest petit s'hagi de tenir en compte a nivell global. Tot i així, en l'àmbit d'aquest projecte, l'impacte és negligible.

Pel que fa a l'impressora, el major impacte es degut al consum de paper per a imprimir [9], en aquest cas, s'ha utilitzat paper reciclat per reduir-lo.

També cal considerar l'impacte ambiental positiu que pot arribar a tenir el desenvolupament d'aquest projecte. La digitalització de documents comporta beneficis per al medi ambient com la reducció d'espai físic per emmagatzemar documentació.

D'altra banda, també comporta un impacte en la societat ja que es gaudiria de més fàcil accés i control de la informació. A més d'un estalvi de temps i recursos per digitalitzar documents manuscrits.



## Conclusions

Aquest projecte ha servit per poder continuar els estudis realitzats pel José de la Puente i la Judith Bergadà, en els quals s'analitzava l'aplicació d'aquest algoritme i es van arribar a aconseguir taxes d'encert del 90%.

En aquest cas, s'ha anat un pas més enllà i s'ha dissenyat algoritme per a la segmentació i reconeixement de lletres escrites de forma conjunta continuant amb els mateixos fonaments aplicats en els anteriors projectes.

S'han determinat les característiques i condicions de contorn que provoquen que una mateixa lletra tingui diferents representacions per tal de tenir-ho en compte alhora de la creació d'un catàleg representatiu. A més, per la creació d'aquest catàleg s'ha tingut en compte la freqüència d'aparició de les lletres en l'idioma a detectar, reduint la informació necessària al catàleg i millorant la taxa de reconeixement.

També, s'ha tingut en compte el paràmetre adimensional de la relació entre la amplada i l'altura de les lletres. Aquest paràmetre és màxim quan correspon a lletres tipus la "m" i permet delimitar la zona de la paraula que conté la següent lletra. També en el cas contrari, quan és mínim (lletres tipus "f") permet determinar a partir de quin punt s'inicia la cerca de la següent lletra.

Finalment, evaluant la taxa d'encert amb la distància de Levenshtein s'ha obtingut una taxa d'encert al voltant del 70%.

Cal remarcar que un error en la identificació d'una lletra en aquest algoritme comporta l'encadenament d'errors posteriors. És per aquest motiu que es valora aquest 70% com a resultat satisfactori.

Per tant, la conclusió d'aquest treball és que s'han complert els objectius inicialment marcats ja que s'ha aconseguit eliminar la restricció de l'anterior projecte disminuint un solament un 20% la taxa d'encert.

Així, es recomana continuar endavant aquest projecte per incloure signes de puntuació, accents i augmentar la velocitat de càlcul.



## Agraïments

En aquest apartat m'agradaria mencionar les persones que han participat en el desenvolupament del treball.

Principalment, el tutor d'aquest projecte per dos motius. En primer lloc, per donar-me l'oportunitat de desenvolupar-lo. En segon lloc, donar les gràcies per la seva gran labor i ajuda en la realització del treball. El tutor ha estat sempre obert a resoldre tots els problemes sorgits, sense la seva experiència i saber aquest treball no hauria estat possible.

També mencionar a les dues persones que han desenvolupat els treballs previs a aquest: la Judith Bergadà i el José Maria de la Puente. L'estudi previ ha estat de gran ajuda per continuar investigant el reconeixement de caràcters.





## Bibliografia

- [1] Judith Bergadà Pijuan. Reconeixement de text mitjançant la SVD. Treball de fi de grau, UPC, Barcelona, 2016.
- [2] José María De La Puente Puigjaner. Reconeixement de text manuscrit mitjançant la SVD. Treball de fi de grau, UPC, Barcelona, 2016.
- [3] Santosh Kumar Henge and B. Rama. Comprative study with analysis of OCR algorithms and invention analysis of character recognition approched methodologies. In *2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEI-CES)*. IEEE, jul 2016.
- [4] Karen Hao. La IA empieza a centrarse en la causas en lugar de las correlaciones. "<https://www.technologyreview.es/s/11156/la-ia-empieza-centrarse-en-la-causas-en-lugar-de-las-correlaciones/>". "[En línea; Data de consulta: 06/06/2019]".
- [5] G. Tauschek. Reading machine. United States Patent 2 026 329, 1935.
- [6] S. Mori, C.Y. Suen, and K. Yamamoto. Historical review of OCR research and development. *Proceedings of the IEEE*, 80(7):1029–1058, July 1992.
- [7] Gonzalo Navarro. A guided tour to approximate string matching. *ACM Computing Surveys*, 33(1):31–88, 2001.
- [8] Gintautas Grigas and Anita Juškevičienė. Letter frequency analysis of languages using latin alphabet. *International Linguistics Research*, 1(1):18, 2018.
- [9] Katarzyna Grzesik and Tomasz Terefeńko. Life cycle assessment of an inkjet printer. *Polish Journal of Environmental Studies*, vol 21:95–105, 08 2012.